MetaFrame XPe Feature Release 2 Enterprise Distribution

By Citrix Consulting Services

Citrix Systems, Inc.

Notice

The information in this publication is subject to change without notice.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CITRIX SYSTEMS, INC. ("CITRIX"), SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR DIRECT, INCIDENTAL, CONSEQUENTIAL OR ANY OTHER DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS PUBLICATION, EVEN IF CITRIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

The exclusive warranty for Citrix products, if any, is stated in the product documentation accompanying such products. Citrix does not warrant products other than its own.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Version History		
1.0	Daniel J. Feller	June 13, 2002
2.0	Jo Harder	December 2002

Table of Contents

Overview

With Citrix achieving the Microsoft Windows 2000 Server certification, Citrix software installations, starting with Feature Release 2, will now follow the installation standards put forth by Microsoft. The installation standards are based around the Microsoft Windows Installer Service. The Windows Installer service is a default service that is configured with the installation of Windows 2000 Server, Windows 2000 Advanced Server, and Windows 2000 Datacenter.

In order to understand what the new installation standards mean for Citrix implementations, one must briefly understand what is incorporated with a Windows Installer Service. This document tries to detail the following:

- The components of the Windows Installer Service
- MetaFrame XPe Feature Release 2 Enterprise Distribution Examples

Windows Installer Service

The installation of MetaFrame XP Feature Release 2 is based off of the Windows Installer Service. In order to understand the procedures necessary to successfully implement Feature Release 2, one should be familiar with the functionality of the Windows Installer Service. This service comprises of the following two mandatory components:

- Client-Side Installer Service (Msiexec.exe)
- Package File (.msi file)

In addition to these two mandatory components, the Windows Installer Service can also be influenced by the following two optional components:

- Transform Files (.mst file)
- Command Line Options

Client-Side Installer Service

The Client-Side Installer Service is a preinstalled operating system service for Windows 2000. The service allows the operating system to manage the entire installation process. Upon execution of a Package File, the Installer Service calls the Installer program, Msiexec.exe. This program utilizes a dynamic link library, Msi.dll, to read the package files, apply transforms, and incorporate command-line options. The Installer Program performs all installation-related tasks including:

- Copying files to the hard disks
- Making registry modifications
- Creating desktop shortcuts
- Displaying dialog boxes for user preferences

When the Windows Installer service is installed on a system, the .MSI file type is associated with the Installer Service. Once the .MSI file is executed, the operating system associates the .MSI file with the Windows Installer and executes the Msiexec.exe application.

Windows Installer must be upgraded to Version 2.0 in order to support the distribution of FR2. The file to upgrade to Version 2.0 is included on the MetaFrame CD and is located as follows:

\MetaFrame\support\msi20\instmsiw.exe

Installation package file

Inside each MSI file is a relational database that stores the installation and un-installation instructions for the program. By using a MSI reader tool from Microsoft, the contents of the relational database can be viewed and modified. The instructions contained within the database have the ability to manage the installation across numerous scenarios. The tables that comprise of the relational database contain the following information:

- Available features
- Components

- Relationships between features and components
- Necessary registry settings

Making modifications to the relational database's tables can be quite cumbersome and difficult. Because of this difficultly, Transform files can be used to make the modifications easier.

Permissions

Executing a Windows Installer package requires that the correct permissions are enabled. Particularly if Terminal Services was installed using only Active Directory permissions, ensure that Group Policies will not block the execution of the Windows Installer file. Within Active Directory, policies that are applied within the Computer Configuration\Administrative Templates\Windows Components\Windows Installer are not configured by default. If a domain-based account is used to invoke the Windows Installer file, lack of sufficient permissions may prevent the proper installation

Transform Files

If the installation package requires modification prior to installation, a transform file (.mst) can be applied. The transform file makes changes to the relational database upon initial installation, manipulating the installation behavior. Once installed, the transform file is cached on the local system. This file is subsequently applied to the base Windows Installer package whenever the Installer needs to make a modification to it.

Transform files can be found in Office 2000 Installation and MetaFrame XPe Feature Release 2 installations, to name a few. Different transform files, included with Feature Release 2, can modify the installation to create a new data store using Oracle or the transform can be used to join an existing MetaFrame Farm with SQL Server as the data store database. Once the transform file is applied to a Feature Release 2 installation, any subsequent installation requests will retrieve the cached copy of the transform file to reapply the configuration settings, in case the installation update modifies the settings.

Transform files can be modified, but require the use of a commercially available tool. However, once the transform file is modified to suit the needs for the installation, execution requires the use of a command line like the following:

msiexec.exe /a package TRANSFORMS=transformsfile

Command Line Options

The Msiexec.exe executable is setup to accept numerous command line parameters. The following table lists the most common command line options and their purpose. For complete information, please read the Microsoft Article: http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winxppro/proddocs/msiexec.asp

Parameter	Description	Example
/i	Installs or configures a package	Msiexec **/I** mfxp001.msi
/a	Applies an administrative install to the package	Msiexec **/a** mfxp001.msi
/f **[p]** **[o]** **[e]**	**/fp** Reinstalls only if file is missing **/fo** Reinstalls if file is missing or if an older version is installed.	Msiexec **/fpecms** mfxp001.msi

Parameter	Description	Example
[d] **[c]** **[a]** **[u]** **[m]** **[s]** **[v]**	**/fe** Reinstalls if file is missing or an equal or older version is installed. **/fd** Reinstalls if file is missing or a different version is installed. **/fc** Reinstalls if file is missing or the stored checksum does not match the calculated value. **/fa** Forces all files to be reinstalled. **/fu** Rewrite all required user-specific registry entries. **/fm** Rewrites all required computer-specific registry entries. **/fs** Overwrites all existing shortcuts. **/fv** Runs from source and re-caches the local package	
/x	Uninstalls the package	Msiexec **/x** mfxp001.msi
TRANSFORMS	Installs a transform file to the installation package. Please note that TRANSFORMS must be in all caps.	Msiexec /I mfxp001.msi **TRANSFORMS**=Example.mst
/q **[n]** **[b]** **[r]** **[f]** **[n+]** **[b+]** **[b-]**	**/qn** Displays no user interface. **/qb** Displays a basic user interface. **/qr** Displays a reduced user interface with a modal dialog box displayed at the end of the installation. **/qf** Displays the full user interface with a modal dialog box displayed at the end. **/qn+** Displays no user interface, except for a modal dialog box	msiexec **/qb** Example.msi

Parameter	Description	Example
	displayed at the end. **/qb+** Displays a basic user interface with a modal dialog box displayed at the end. **/qb-** Displays a basic user interface with no modal dialog boxes	

Many of the aforementioned options can be added to each other for an installation. For example, the following command line will install the MFXP001.msi file with a RemoteDB.mst transform added while installing with no user interface.

msiexec.exe /a mfxp001.msi TRANSFORMS=RemoteDB.mst /qn

Process Flow

Based on the above descriptions of the components comprising of an MSI installation, the following diagram tries to graphically depict this process.

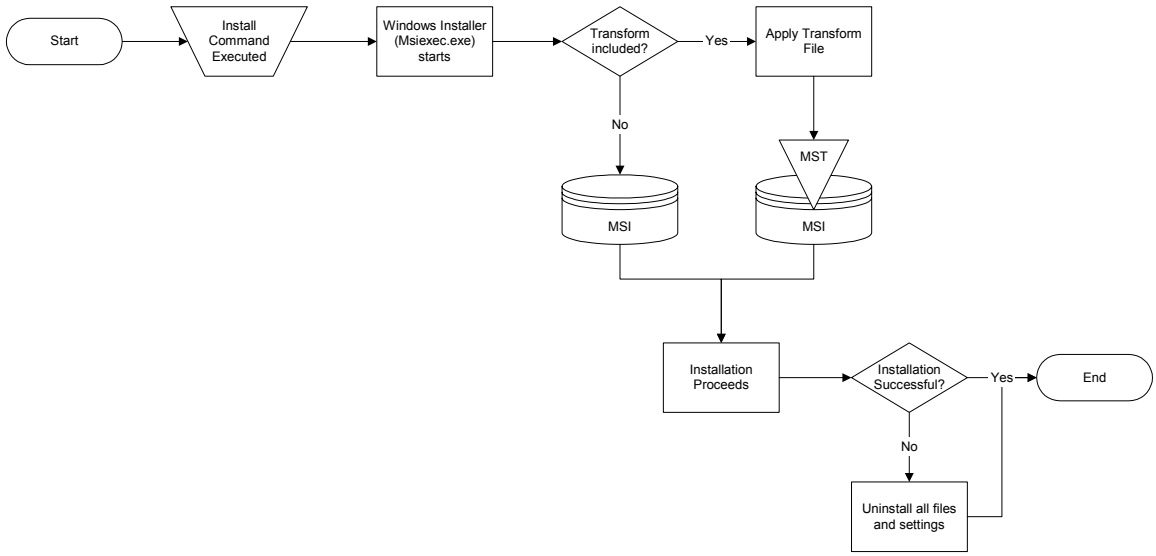

Figure 1: Windows Installer Process Flow

- The installation procedure begins by the user executing the installation command.
- Once done, the Windows Installer Service (Msiexec.exe) is executed because of the file type association between MSI files and the Msiexec.exe executable.
- If a transform file (.MST) was associated with the executed installation command, the Windows Installer Service will integrate the MST file with the MSI file before the installation begins.
- Installation proceeds with modifications being made to files, registry entries, and shortcut creation.

- If the installation was successful, the process ends. However, if the installation failed, the Windows installer server will undo every installation modification just done.

MetaFrame XPe Feature Release 2 Installation Options

Once the functionality of the Windows Installer Service is understood, one can better understand how all the components function together to provide for a successful implementation environment. Feature Release 2 is configured with three different installation types. They include:

- Manual Installation
- Silent Installation with an Answer File
- Silent Installation with a Transform File

Manual Installation

The simplest installation method to use is the manual installation. This is simply accomplished by inserting the Feature Release 2 CD and executing the MFXP001.MSI file. The installation starts, with the help of the Windows Installer Service. The administrator is prompted with numerous questions in order to properly configure and install Feature Release 2.

Although this is the easiest method, it is not the most scalable and efficient. This method is extremely difficult to do in a large-scale enterprise environment, especially for an environment where the servers are not all located in a single data center.

Also, because of the manual process of this method, discrepancies are bound to introduce themselves into the environment. These subtle differences can manifest themselves into strange and unique issues with the farm at a later date.

Because of these issues, this is not a recommended method for deploying Feature Release 2 to enterprise clients.

Silent Installation with an Answer File

The second installation method for Feature Release 2 is an unattended installation with the use of an answer file. The answer file is a preconfigured file that is used in conjunction with an installation executable that looks for configuration answers from the answer file. The template answer file, located in the \Support\Install CD directory, can be modified to any environment configuration.

Please note that this method requires that a local administrator password be exposed in clear text. This may present a security breach that may be unacceptable for some companies. Further, customizations of the answer file are limited and thus may not incorporate all of the specific settings desired.

This method is an excellent practice when installing new systems, especially if the new systems are being created with an unattended server build script. The installation of Feature Release 2 can simply be integrated into the unattended server build script with a simple command:

UnattendedInstallation.exe <MSI File Name> <Answer File>

This method, however, encounters difficulties when an environment is already built and functioning. Within this environment, this method requires a script to be executed upon every server in order to install Feature Release 2. This poses some challenges like:

- How does one call the unattended install script?
- Can the installations be controlled centrally?

- Is there feedback as to the success/failure of the install?

These three questions show why it is difficult to use this method for a production environment.

Silent Installation with a Transform File

The final installation method included with Feature Release 2 is the Silent Installation with a Transform File. Citrix has made the Feature Release 2 of the MetaFrame XPe product a Windows Installer Service installation that can be distributed with the use the Citrix Installation Manager. The MSI file, MFXP001.msi, for Feature Release 2 can be found on the MetaFrame XP Feature Release 2 CD under the \MF Directory.

Citrix has also included a number of helper transform files to assist in the installation of Feature Release 2. These transform files can be found on the MetaFrame XP Feature Release 2 CD under the \Support\Install directory. Once modified with a transform file editor, the transform files provide for the following environments:

- IgnoreMSICheck.mst – A transform file that is used to override the installer's check for MSI Version 2.0.
- Join_Indirect.mst – A transform file that is used to join a MetaFrame farm connecting to the data store configured for indirect mode.
- LocalDB_Access_Create.mst – A transform file that is used to create a new farm with a data store that is created locally with Access.
- ThirdPartyDB_Create_Direct.mst – A transform file that is used to create a new farm with the data store located on a different server but configured for direct mode.
- ThirdPartyDB_Join_Direct.mst – A transform file that is used to join an existing farm with a data store that is installed on a different server but configured for direct mode.

Editing of the transform file can easily be accomplished by an editor such as ORCA, which may be downloaded free of charge from http://www.microsoft.com/msdopwnload/platformsdk/sdkupdate.

Once the appropriate transform file is modified for environments that already have MetaFrame XPe deployed, Installation Manager can be used to successfully deploy Feature Release 2 to the entire MetaFrame XPe farm. This process answers the three questions that made the Silent Installation with an Answer File difficult for a production environment.

- Installation Manager controls the script execution on every server
- Installation Manager is controlled from a central location via the Citrix Management Console
- Installation Manager provides success and failure feedback within the Citrix Management Console

For environments that do not have MetaFrame XPe deployed, the same process can be used, provided that the correct modifications are made to the transform file. Installation Manager and Resource Manager are installed by default.

Enterprise Deployment by Automated Build

Many enterprise environments will start to deploy MetaFrame XPe with the release of Feature Release 2. With no current MetaFrame environment in existence, the previous deployment example is not valid. The following example describes the process needed to deploy MetaFrame XPe Feature Release 2 to a new environment. This example only describes how to automate the Feature Release 2 part of the automated build. This description can be integrated with a complete server automated build to provide a complete solution.

Environment Description

The environment consists of the following characteristics:

- A single MetaFrame XPe server that was manually created with Feature Release 2
- A single Microsoft SQL Server 2000 Data Store (SQLDS) located in Data Center A
- Domain Name: US_LAB
- Data Store database name: MetaFrame_XPe_DS
- Data Store DB Owner Username: DSAdmin
- Data Store DB Owner Password: DSPwd

Constraints

- Administrator does not want to manually install the servers by hand

Implementation Steps

By following the remaining sections, the environment can easily be upgraded to MetaFrame XPe Feature Release 2 from a central location.

Network Location Creation

- The administrator should create a network share on a file server, which is located in Data Center A.
    - Share A: \\FileServerA\InstallFiles
- All MetaFrame servers, within the same data center location, should have access to the local network share.
- In order to correctly populate the network share with the appropriate installation files, an administrative installation must occur on each network share. This can easily be accomplished by executing the following command: **Msiexec /a mfxp001.msi**
- Once the command is entered, the installation will begin by requesting the network location where to store the files. This location should be the same location that was created in the initial step.

- \\FileServerA\InstallFiles for Data Center A

- Copy, from the Feature Release 2 CD, the following files and place in the same directory as the administrative install:
  - UnattendedInstall.exe
  - XPFR2_UnattendedTemplate.txt

Unattended Text File Creation

- Edit the XPFR2_UnattendedTemplate.txt file so it resembles the following[1]:

```
[MetaFrame License Agreement]

AcceptLicense=Yes


[Data Store Configuration]

CreateFarm=No

DirectConnect=Yes


[Direct Connect Settings]

DSNFilePath=C:\Program Files\Citrix\Independent Management
Architecture\MF20.dsn

UserName=US_LAB\DSAdmin

Password=DSPwd


[Shadowing Restrictions]

AllowShadowing=Yes

ProhibitRemoteControl=No

ProhibitNotificationOff=No

ProhibitLoggingOff=No


[Citrix XML Service]

ExtendIIS=No
```

[1] The unattended text file will be different based on the environment.

```
; This setting applies only if ExtendIIS is No
DedicatedPortNumber=82

[NFuse]
SetDefaultPage=Yes

[Update ICA Clients]
UpdateClients=No
ClientPath=

[Options]
RebootOnFinish=Yes
LogFile=c:\msi.log
UILevel= BASIC_UI_NO_MODAL
FeatureRelease=Yes

[MetaframeServer]
ServerType=Metaframe Enterprise Server
```

- Save the file as XPFR2Unattend.txt

Unattended Installation of Feature Release 2

- Log on to the server as an administrator and open a command prompt.
- Type in the following commands:

  net use z: \\FileServer\InstallFiles

  Z:

  cd mfxpfr2

  UnattendedInstall.exe MFXP001.msi XPFR2Unattend.txt
- Once complete, the server will install MetaFrame XPe Feature Release 2.

Enterprise Deployment by Installation Manager

Most large scale, enterprise environments are going to require a simple and centralized way to distribute updates to the entire MetaFrame Farm. The sections that follow tries to create a realistic enterprise environment giving the steps needed to successfully distribute the latest Feature Release for MetaFrame XPe.

Environment Description

The environment consists of the following characteristics:

- 200 MetaFrame XPe Feature Release 1 servers
- 100 servers are located in one Data Center A with the remaining 100 servers located in a second Data Center B
- A single Microsoft SQL Server 2000 Data Store (SQLDS) located in Data Center A
- Each server has the Installation Manager service installed
- Data Store database name: MetaFrame_XPe_DS
- Data Store DB Owner Username: DSAdmin
- Data Store DB Owner Password: DSPwd

Constraints

- Administrator does not want to manually sit in front of each server installing the Feature Release.
- Installations should not occur over the WAN link connecting the two data centers

Implementation Steps

By following the remaining sections, the environment can easily be upgraded to MetaFrame XPe Feature Release 2 from a central location.

Network Location Creation

- The administrator should create two identical network shares on two separate file servers, with one file server in Data Center A and the other file server in Data Center B.
    - Share A: \\FileServerA\InstallFiles
    - Share B: \\FileServerB\InstallFiles
- All MetaFrame servers, within the same data center location, should have access to the local network share.
- In order to correctly populate the network share with the appropriate installation files, an administrative installation must occur on each network share. This can be accomplished by executing the following command: **Msiexec /a mfxp001.msi**

- Once the command is entered, the installation will begin by requesting the network location where to store the files. This location should be the same location that was created in the initial step.
    - \\FileServerA\InstallFiles for Data Center A
    - \\FileServerB\InstallFiles for Data Center B

Transform File Creation

- Download the Windows Installer SDK from Microsoft and at the following address: http://www.microsoft.com/msdownload/platformsdk/sdkupdate/
- Start the ORCA application, part of the Windows Installer DSK, and load a backup copy of the Feature Release 2 MSI file (MFXP001.MSI)
- Select **Transform → Apply Transform…** from the Menu Bar

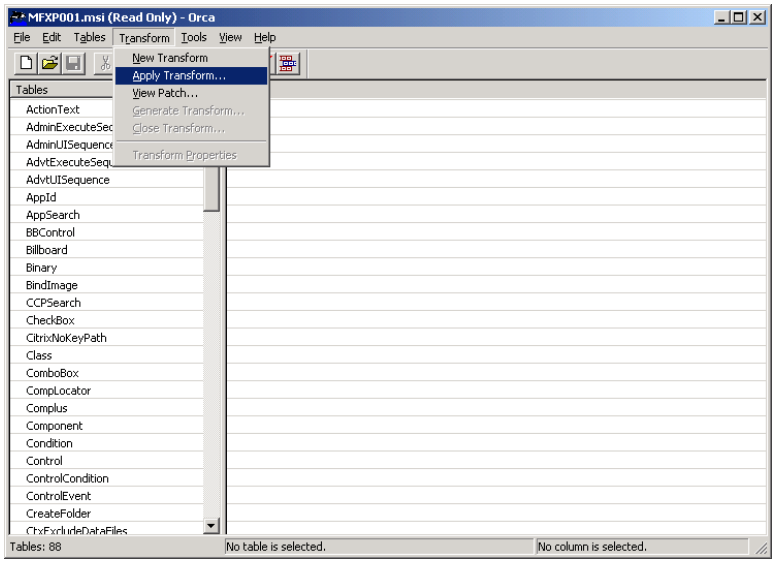

Figure 2: Apply Transform

- Select the appropriate transform file from the five on the Feature Release 2 CD in the \Support\Install directory. For this configuration the ThirdPartyDB_Join_Direct.mst file should be used.
- Once the transform file is applied to the MSI file, a table on the left pane should have a green hash mark. Upon selecting this table, the properties appear in the right pane with the modified settings appearing with a green border.

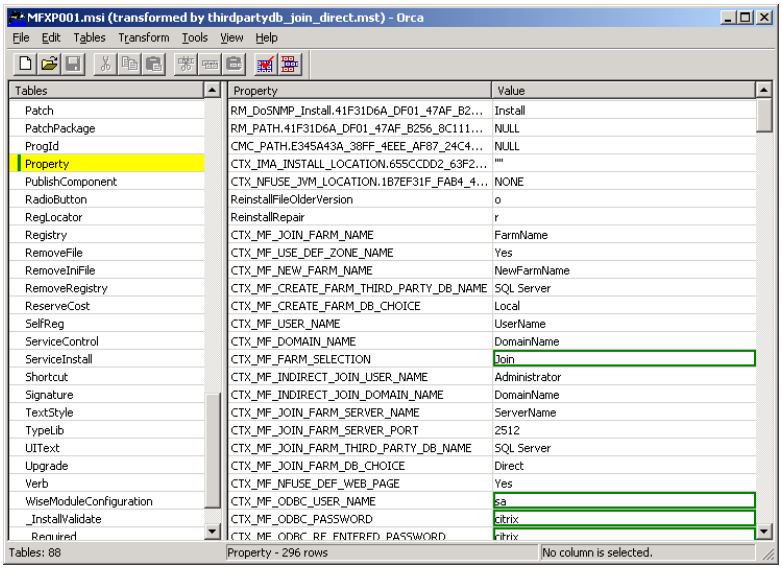

Figure 3: Modify Transform

- Modify the CTX_MF_ODBC_USER_NAME entry to **DBAdmin**
- Modify the CTX_MF_ODBC_PASSWORD entry to **DBPwd**
- Modify the CTX_MF_ODBC_RE_ENTERED_PASSWORD entry to **DBPwd**
- Modify the CTX_MF_SILENT_DSNFILE entry to **C:\Program Files\Citrix\Independent Management Architecture\MF20.dsn**
- Modify the CTX_MF_LAUNCH_CLIENT_CD_WIZARD entry to **No**
- On the Menu Bar select **Transform → Generate Transform…**
- Supply a new transform file name and save to the same path as the administrative install for MetaFrame XPe Feature Release 2. The transform file should also be copied to the second data center's network share.

Installation Manager Package

- Launch the Citrix Management Console (CMC) and access the Installation Manager Plug-in[2].
- Create a new Installation Manager Package. Enter in a Package Name and enter in the path to the administrative install created earlier. The path should be in a UNC format (\\servername\sharename). Once added, select **OK**.

[2] This procedure is for a MetaFrame server in Data Center A. The same procedure should be replicated to a server located in Data Center B.

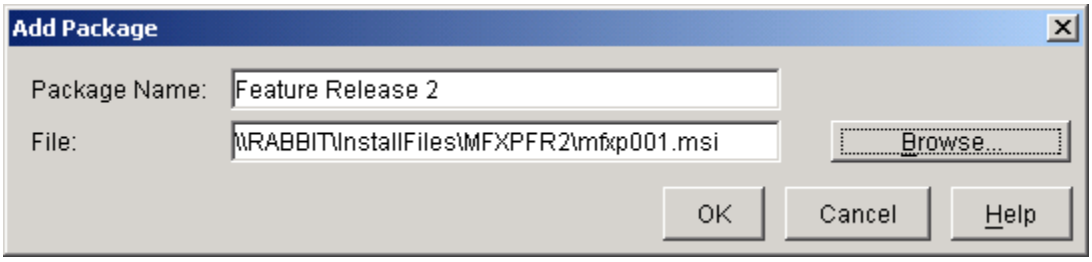

Figure 4: Add Package Window

- Click **Yes** to add a transform file and command options to the package
- Click **Add** and select the transform file created earlier[3].

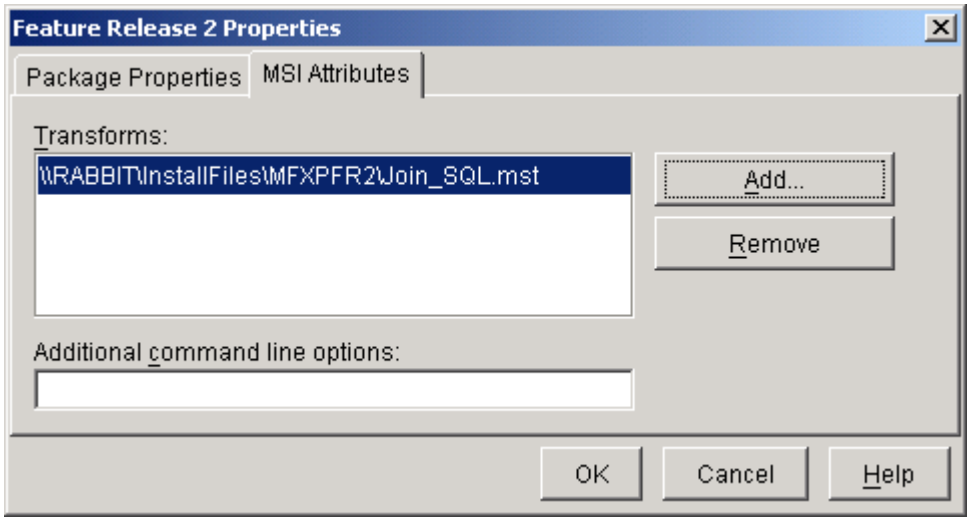

Figure 5: Add Transform Window

- Select the package and select from the Menu Bar: **Actions** → **Installation Manager** → **Install Package**.
- Select the servers to install Feature Release 2 to. For the enterprise deployment, this should be limited to 10-20 servers at a time[4]. Once the servers are selected, select **Next**.
- Select the appropriate scheduling for the installation and select **Finish**.
- Installation Manager will distribute the package to the appropriate servers. Once complete, those servers will contain Feature Release 2. Repeat process for remaining servers and data centers.

---

[3] Transform file should be in the same directory as the MSI package for Feature Release 2.

[4] Feature Release 2 for MetaFrame XPe will automatically force the server to reboot upon completion. Ensure that no users are connected prior to deployment of Feature Release 2.

Summary

The key concept for MSI file installations is that all installations based on the MSI format contain, at a minimum, two components:

- Client-Side Installer Service (Msiexec.exe)
- Package File (.msi file)

Provided that the client Windows Installer is updated to version 2.0 and sufficient permissions are allocated within Active Directory (if required), using the Windows Installer package of Feature Release 2 is a fast and efficient means of deploying both base MetaFrame XP installations and upgrades. Further, by following an automated process, the deployments will be consistent.

Having the ability to create transform files and applying them to the installation package can simplify the installation process. Using an enterprise software distribution tool, like Installation Manager, can greatly reduce the time needed to push out new MSI installations to the MetaFrame XPe environment.

The two different examples are only a brief glimpse as to how the distribution of Feature Releases can be made easier with the correct tools and the knowledge to use the tools to their potential. By following these steps, enterprise deployments can be made easier, more efficient, and have a greater amount of control.

851 West Cypress Creek Road Fort Lauderdale, FL 33309 954-267-3000 http://www.citrix.com